

Entrenamiento General para Front end #2

Fecha última actualización

08/20/2013

Versión

v1.00

1 Historia del Documento

Fecha	Descripción del Cambio	Autor	Versión
08/20/2013	Creación	rnunez	v1.0
06/11/2014	Modificación	Dballesteros	v1.1

2 Resumen

El presente es el plan y programa de entrenamiento en diversos temas de calidad y corrección de código orientado mayormente en desarrollo para frontend y cual intenta cubrir ciertos temas de actualización, herramientas, estándares y buenas prácticas de desarrollo de este particular.

Éste entrenamiento va enfocando en generar conocimiento práctico de nivel medio en diversos temas relacionados con el desarrollo de software para frontend y en general para fortalecer las buenas practicas de desarrollo de la compañía y a nivel personal profesional.

3 Metodología

Este training va a seguir la siguiente metodología

- Asignar un tema diferente por semana a máximo 2 semanas según el tema
- Cada persona va a dedicarle con tiempo de la empresa de 2 a 3 horas de estudio y práctica
- En forma grupal e individual se harán practicas que van relacionadas en todo lo posible con las tareas y trabajo actualmente asignado
- Al finalizar las tareas serán vistas en grupo o grupos para su posterior análisis y retroalimentación

Objetivo

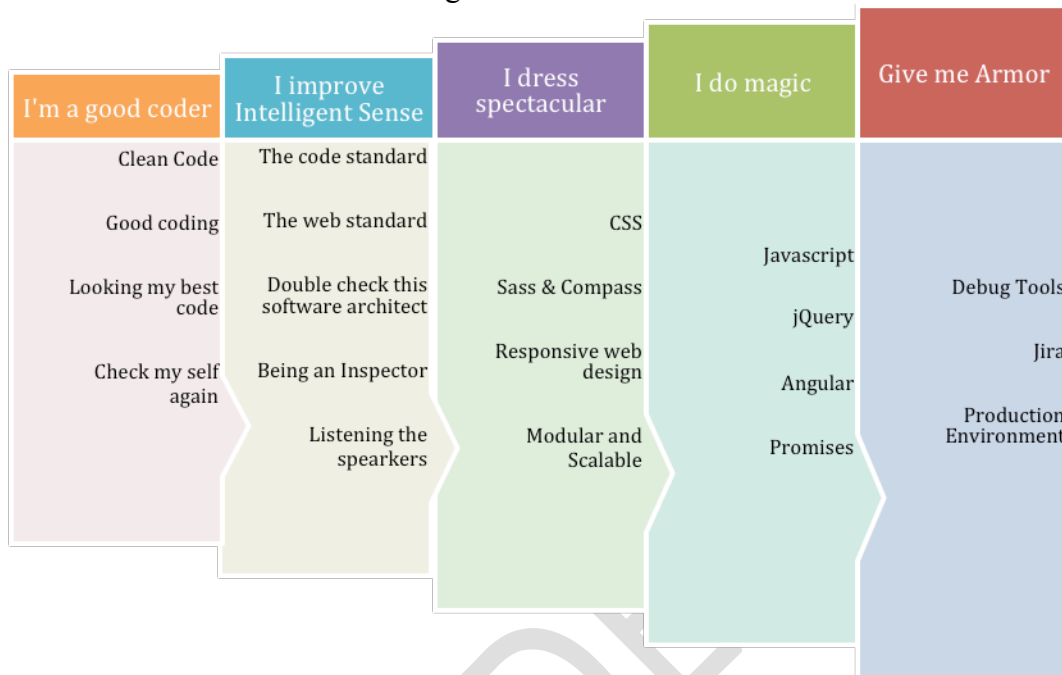
Realizar un aprendizaje tipo practica por medio de la experiencia y objetivos específicos en diversos temas relacionados a la ingeniería de software y el desarrollo de soluciones para front end. Además de fomentar la calidad del trabajo y código a nivel personal, grupal y empresarial creando un perfil mucho más competitivo y dinámico.

Evaluación

La evaluación no busca medir los resultados del training en términos de bueno o malo, más bien, todo lo contrario la evaluación buscará cumplir objetivos de aprendizaje propios de cada persona, pues cada persona posee sus propias formas creativas de aprender y aplicar, por ende será el aprendizaje a través de la aplicación lograda lo que determinará la evaluación de cada training.

4 Temas

La lista de temas a evaluar en éste training son:



I'm good coder

Escribir buen código distingue a un desarrollador, conocer como debe ser un buen código adelanta el éxito de un líder técnico, confiar en la calidad del código permite al arquitecto de software tomar mejores decisiones y soluciones; un excelente código en general en todas las soluciones se traduce en planeación de producto, nuevos negocios, éxito y mejores ingresos para un administrador.

La calidad de código requiere de conocer buenas practicas, practicar incesantemente, utilizar herramientas de apoyo y estar sujeto a revisión y mejoras constantes. La calidad de código permite escribir rutinas mucho más fácil, con menos esfuerzo y con más claridad.

El objetivo de ésta primer unidad es repasar esas practicas para aprender de ellas y tenerlas presente, aplicarlas y mantenerlas en practica constante.

Lecturas:**Clean_code.pdf (usando los números de página impresos):****Localización:** <https://drive.google.com/open?id=0B53rte11kwzUX2dNa3ZHeTNiaGc>

- Functions, páginas 31-36
- Formating, páginas 75-84
- Classes, páginas 135-141

Code Complete Second Edition**Localización:** <https://drive.google.com/open?id=0B53rte11kwzUUkpYZmh4RG4zV0k>

- Working Classes, Capitulo 6, página 3-5, 10-12, 16-18, 31-34
- High Quality Routines, Capitulo 7, páginas 1-6
- Defensive Programming, Capítulo 8, páginas 2-5, 9-11
- Controlling Loops, Capítulo 16, páginas 1-3, 21-23

Asignaciones :

- Seleccione código que usted haya desarrollado en los últimos meses o semanas
- Extraiga 5 ejemplos diferentes donde haya aplicado las buenas practicas de codificación estudiadas en las lecturas
- Extraiga 5 ejemplos de casos diferentes donde no haya aplicado alguna de las buenas practicas de programación estudiadas y haga la corrección de tal forma que quede documentado como estaba el código antes y como quedó después del arreglo
- Los ejemplos serán vistos en forma colectiva en una sesión de trabajo con todos los compañeros
- Ésta asignación tiene una duración de una semana

I Improve Intelligent Sense

“Every line of code should appear to be written by a single person, no matter the number of contributors.”

Lecturas:**Standard de Pixel**<https://gist.github.com/davibq/27689c782256c5316cec>**Standard de Airbnb**<https://github.com/airbnb/javascript/tree/master/es5>**Code Guide****Localización:** <http://mdo.github.io/code-guide/>**Autor:** Mark Otto (Bootstrap creator)

Asignaciones :

- Estudiar el web standard que esta adjunto
- Descargar Yeoman y crear un proyecto con el generador webapp. Asegurarse que todo funciona correctamente.

I Dress Spectacular

¿Que hay de nuevo en HTML5?

En la siguiente presentación se repasan las nuevas características que se presentan en HTML5:

<http://www.slideshare.net/maxdesign/a-quick-introduction-to-html5>

CSS

En caso que nunca haya trabajado con CSS, ver el siguiente video

Styling Your First HTML5 Web Page with CSS3

Localización:

<http://channel9.msdn.com/Series/HTML5-CSS3-Fundamentals-Development-for-Absolute-Beginners/Styling-Your-First-HTML5-Web-Page-with-CSS3-03>

Revisar la siguiente guía de conceptos básicos de CSS: <http://adamswartz.co/magic-of-css/>

Ejercicio CSS:

Utilizando únicamente CSS y HTML, cree una página como la siguiente:



- Las fuentes de los textos deben ser diferentes y al menos una debe utilizar un Google Font (<https://www.google.com/fonts>)
- En vez del círculo, la estrella y el hexágono debe colocar un ícono de la red social a la que hace referencia el texto. Para hacerlo, utilice este sprite <http://brandonsetter.com/demos/super-massive-css-sprite-social-icon-set/supermassivecssspriteicons2.png> y el sitio web <http://www.spritecow.com/> para saber las coordenadas de cada ícono. El ícono debe cambiar cuando el usuario ponga el mouse encima y debe existir un link en los íconos hacia la red social.

Media Queries

Los media queries son utilizados en CSS para agregar estilos dependiendo de una serie de reglas que el dispositivo debe cumplir.

Por ejemplo, podemos realizar un media query que agregue los estilos contenidos en este únicamente para dispositivos con un ancho entre los 320px y los 480px.

En este enlace se explica como implementar los media query: <http://cssmediaqueries.com/what-are-css-media-queries.html>

En este link se encuentran predefinidos algunos media queries utilizados muy comúnmente: <http://css-tricks.com/snippets/css/media-queries-for-standard-devices/>

Responsive Web Design

Un sitio web responsive permite que la página web se adapte a las dimensiones del dispositivo en el que es visualizada.

El diseño responsive puede realizarse manualmente utilizando media queries, o bien mediante frameworks css que proveen sistemas de grid para realizar el responsive web design de una manera más sencilla.

Como ejemplos de los sistemas de grid tenemos el de Pure y el de Bootstrap. Que se pueden observar en las siguientes direcciones: <http://getbootstrap.com/css/#grid> y <http://purecss.io/grids/>. Ambos funcionan de manera similar.

La idea principal detrás de los sistemas de grid es dividir la pantalla en un número determinado de columnas para posteriormente definir cuantas columnas debe ocupar un objeto HTML en cada dispositivo. Estas especificaciones se realizan agregando clases css a los objetos.

Revisar los siguientes enlaces para entender el funcionamiento de los sistemas de grid:

- <http://scotch.io/bar-talk/understanding-the-bootstrap-3-grid-system>
- <http://www.helloerik.com/bootstrap-3-grid-introduction>

En este vídeo nos muestran la diferencia entre fixed, fluid, adaptative y responsive. Todas son diferentes técnicas para visualizar sitios web en diferentes dimensiones de pantallas:

<http://teamtreehouse.com/library/build-a-responsive-website/introduction-to-responsive-web-design/fixed-fluid-adaptive-and-responsive-2>

Ejercicio Grids y Media Queries

Utilizando el ejercicio de CSS (Redes Sociales). Haga un diseño responsive utilizando algún framework de grids (Pure o Bootstrap). Además en dispositivos móviles, los fondos de los textos de las redes sociales deben ser verde (#27782d), rojo (#f42929) y morado (#782740).

Sass & Compass:

Sass and Compass in Action

Localización: <https://drive.google.com/open?id=0B53rte11kwzUTjg1WmlzTnE4a2c>

- Capítulo 1 "Sass and Compass make stylesheets fun again", páginas 6-12
- Capítulo 2 "Basic Sass syntax, páginas 26-44

* Si desea profundizar en el uso de sass (Expressions, Functions, ...)

- Capítulo 4 "Advanced Sass and Compass"

En caso de que desee el código fuente de este libro, puede descargarlo de la siguiente dirección:

- <https://github.com/pengwynn/sass-and-compass-in-action>

Modular y Escalable:

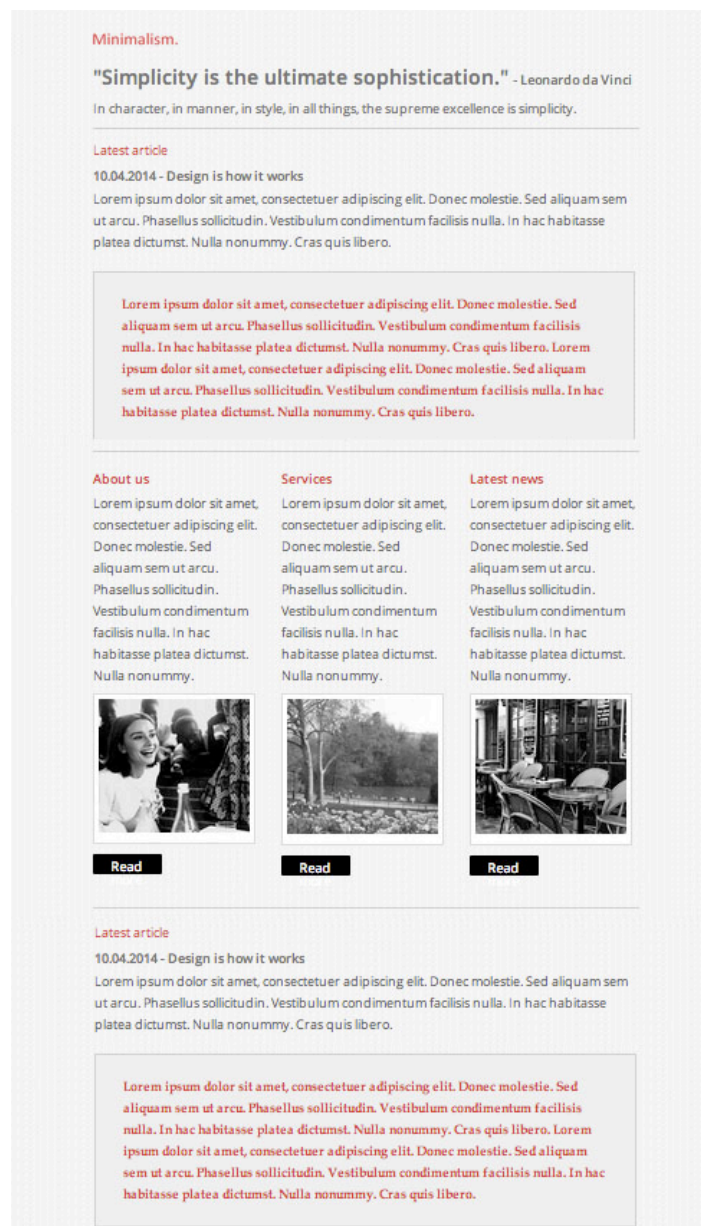
A la hora de escribir CSS no solo debemos preocuparnos porque el sitio se vea de la manera deseada, sino que también debemos tener en cuenta que el CSS debe ser modular y escalable. Para esto existen una serie de metodologías que permiten un CSS escalable, reutilizable y flexible.

Dentro de estas metodologías se encuentra OOCSS (Object Oriented CSS), SMACSS (Scalable and Modular Architecture for CSS), DRY CSS (Don't Repeat Yourself CSS), BEM (Block, Element, Modifier).

A continuación se encuentra una presentación con las ideas principales acerca de OOCSS y ejemplos de cómo implementarlo: <http://www.slideshare.net/stubbornella/object-oriented-css> y <http://www.slideshare.net/DanOlsavsky/oocss-20284331>

Asignaciones:

1. Reproducir el siguiente website http://www.marijazaric.com/minimalism_responsive/index.html
2. Utilizando Sass y Compass como herramientas
3. El sitio tiene que ser funcional para Chrome, Firefox e IE (version 8.0)
4. Utilice OOCSS. Para comprobar su funcionamiento, debe realizar una página adicional igual a la realizada en el punto #1 pero en el siguiente orden (artículo, contenido, artículo). La idea es que no tenga que agregar más CSS para la realización de esta página.
5. Al final de la asignación haga un build con Grunt o Gulp para que el proyecto quede listo para producción.
6. Los colores y fuentes pueden cambiar, pero se debe mantener la idea responsive del sitio.



I do Magic

Esta sección del entrenamiento consiste en conocer a fondo el lenguaje javascript, así como algunas librerías para una mejor utilización.

En el siguiente enlace se presentan los fundamentos de Javascript que deben ser comprendidos en su totalidad antes de continuar con las siguientes lecturas (excepto los puntos 13 y 15):

<https://developer.mozilla.org/es/docs/JavaScript/Guide>

En el siguiente enlace encontramos patrones de diseño muy útiles aplicados a Javascript <http://addyosmani.com/resources/essentialjsdesignpatterns/book/>. Aunque todos los patrones son muy importantes, repase principalmente Module Pattern y Singleton Pattern.

Ejercicio (tomado desde [codewars.com/kata/536e9a7973130a06eb000e9f](https://www.codewars.com/kata/536e9a7973130a06eb000e9f)):

1. Escriba una función llamada damage, que calcule el daño que causa un ataque de un Pokemon a otro. El daño es calculado con la fórmula:

damage = 50 * (attack / deffense) * effectiveness

La efectividad puede ser:

- Super efectiva: 2x daño
- Neutral: 1x daño
- No tan efectivo: 0.5x daño

Y debe apegarse a estas reglas:

- fire > grass
- fire < water
- fire = electric
- water < grass
- water < electric
- grass = electric

Además: un ataque contra Pokemon del mismo tipo no es muy efectivo y las reglas dadas anteriormente son simétricas (grass < fire).

La función acepta como parámetros 2 objetos Pokemon. El primero es el que ataca. El objeto Pokemon se define así:

```
{
  "name": "NOMBRE_POKEMON",
  "tipo": "TIPO_POKEMON",
  "attack": "ATTACK",
  "defense": "DEFENSE"
}
```

El resultado de la función debe ser redondeado.

jQuery

jQuery es una librería para la manipulación del DOM. Es por mucho, la librería mas popular de javascript en la actualidad.

Aunque existen librerías similares a jQuery como Prototype (<http://prototypejs.org/>) o MooTools (<http://mootools.net/>) ninguno ha llegado a tener la popularidad de jQuery.

En el siguiente enlace puede encontrar un tutorial muy completo que comprende todos los conceptos que debe conocer de jQuery antes de seguir adelante:

<http://www.desarrolloweb.com/manuales/manual-jquery.html>

* Lectura opcional: si quiere conocer y entender un poco más a fondo la librería jQuery puede visitar este enlace <http://www.smashingmagazine.com/2014/05/29/mystery-jquery-object-syntax-basic-introduction/>

Adicionalmente se ofrecen algunas recomendaciones importantes en el uso de jQuery:

- Los eventos deben ser registrados dentro de la función document ready o posterior a esta.
- Si asignamos un objeto jQuery a una variable, el nombre de la variable debe comenzar con un \$.
- Los métodos de jQuery retornan objetos jQuery por lo que puede darse un encadenamiento de funciones:

```
$someDiv
.attr('class', 'someClass')
.hide()
.html('new stuff');
```

- jQuery es una herramienta muy útil, pero no hay que olvidar que es un wrapper para las funciones que ya nos provee el navegador:

```
$('#miDiv').width(); es igual a document.getElementById('miDiv').width
```

Finalmente, en esta presentación se encuentra un resumen de buenas prácticas en el uso de jQuery: <http://gregfranko.com/jquery-best-practices/>

AJAX y HandleBars JS

Asynchronous JavaScript and XML (AJAX) es utilizado para hacer llamadas al servidor o servicios externos mediante javascript. La ventaja que introduce AJAX es que para traer información nueva no es necesario realizar postback al servidor ni refrescar la página.

El navegador nos permite hacer estas llamadas mediante el objeto XMLHttpRequest, pero como pudimos ver en los capítulos 52 y siguientes del manual de jQuery, este nos permite realizar las llamadas de una manera más fácil.

En este enlace <http://jsfiddle.net/3GBxw/> encontramos un ejemplo de un llamado a un API externo. En el javascript tenemos código HTML utilizado para insertar el nombre dentro de un <p>. Esta es considerada una mala práctica, ya que el código HTML únicamente debe encontrarse en archivos .html

al igual que el código javascript y css.

Para evitar estas malas prácticas, existen librerías como HandleBars JS (<http://handlebarsjs.com/>) o Mustache (<http://mustache.github.io/>) que permiten el uso de plantillas definidas en el HTML para ser compiladas y posteriormente inyectadas en el HTML donde sea necesario.

A continuación se presenta un enlace donde se explica por que HandleBars y como utilizarlo: <http://www.desarrolloweb.com/manuales/manual-handlebars.html>

El atributo “data”

Muchas veces es necesario incluir atributos con información extra en el código HTML. Estos atributos no influyen en el comportamiento del objeto ni cambian su apariencia pero nos pueden ser muy útiles en javascript para desplegar o almacenar información que necesitaremos posteriormente. En este link se encuentra un ejemplo de cómo utilizar el atributo data: <http://jsfiddle.net/4XuCf/> . Haga click en los nombres de las personas para ver el funcionamiento del ejemplo.

Ejercicio:

Haciendo un llamado al siguiente URL:

<http://api.ht.fuseamplify.com/api/artist/top?aggregate=true>

Construya un ranking basado en el campo “volume”. Cada ítem del ranking debe verse de la siguiente manera:

Figura 1:

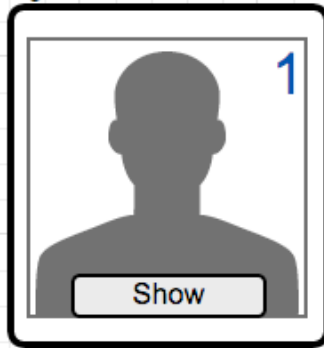
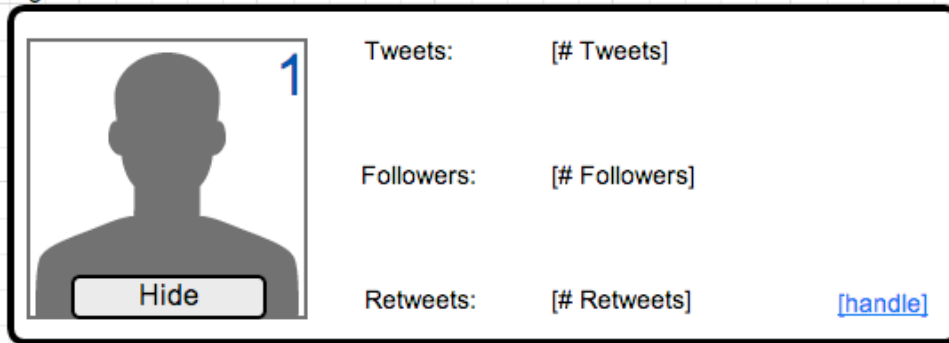


Figura 2:



- El botón “Show” debe mostrar la información como se muestra en la figura 2.
- El botón “Hide” debe esconder la información como se muestra en la figura 1.
- El link “Handle” debe llevar al URL [https://twitter.com/account/redirect_by_id/\[twitter_id\]](https://twitter.com/account/redirect_by_id/[twitter_id])
- El “1” dentro de la imagen indica la posición que ocupa en el ranking.

Debe utilizar jQuery y Handlebars.

Callbacks and Promises

Información acerca del uso de callbacks: <http://www.impressivewebs.com/callback-functions-javascript/>

Como se pudo observar, los callbacks son simplemente llamadas a funciones. Este comportamiento no ocupa de ningún tipo de librerías para su funcionamiento pues es nativo en javascript. Por el contrario, los promises fueron implementados muy recientemente en javascript por lo que su uso recae principalmente en librerías. En este documento se encuentra información acerca del porque utilizar promises y como utilizarlas con la librería Q <http://documentup.com/kriskowal/q/>

Para entender más a fondo porque debemos utilizar callbacks, promises y en general como funciona Javascript por debajo, ver este video: <http://2014.jsconf.eu/speakers/philip-roberts-what-the-heck-is-the-event-loop-anyway.html>

Angular JS

Angular JS es un framework que permite crear Single Page Applications (SPA) en javascript. Esto significa que la página es cargada una sola vez y el contenido se cambia utilizando javascript.

Para una pequeña introducción de Angular JS observar este vídeo:

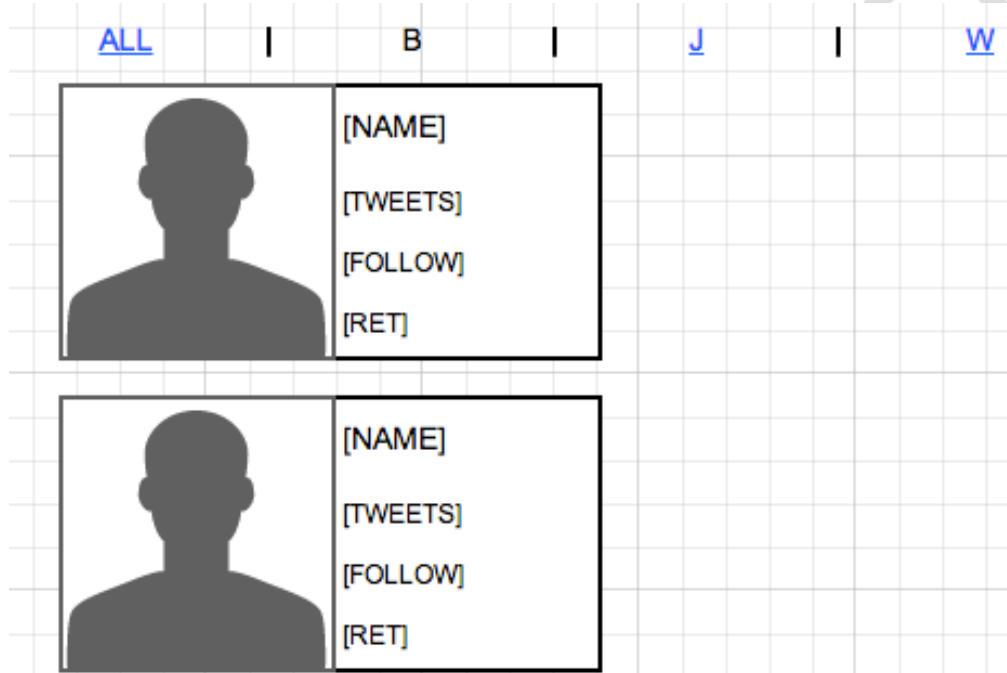
<https://www.youtube.com/watch?v=tnXO-i7944M>

Ejercicio:

Utilizando Angular JS, cargue los datos del siguiente URL:

<http://api.ht.fuseamplify.com/api/artist/top?aggregate=true>

Una vez hecha la llamada al endpoint, agregue un botón para cada letra inicial del nombre del artista y realice un filtrado.



Give me Armor

Chrome Dev Tools

El navegador Google Chrome y sus Dev Tools son una herramienta fundamental para todos los desarrolladores web. Permite trabajar con HTML, CSS y Javascript.

En este enlace se repasan las principales herramientas que nos provee los Dev Tools de Chrome: <https://developer.chrome.com/devtools/docs/dom-and-styles>

Y en este se muestra como sacarle el máximo provecho a los Dev Tools con vídeos y ejercicios: <http://discover-devtools.codeschool.com/>

Jira

Jira es un sistema en línea donde se ingresan tickets que contienen deben ser resueltos por los desarrolladores. Cada ticket tiene la información necesaria para que el desarrollador lleve a cabo la tarea solicitada. En el siguiente vídeo se muestra un poco la herramienta:

<https://www.youtube.com/watch?v=xrCJv0fTyR8>

Herramientas en Producción

Al pasar a un ambiente de producción, el rendimiento del sitio se vuelve muy importante. Existen diferentes sitios con los que podemos medir el rendimiento de un sitio web y a la vez tener sugerencias de que podemos mejorar en nuestro sitio. Un ejemplo es <http://gtmetrix.com>

Para empezar a optimizar el sitio, es importante minificar el javascript y el css. Para optimizar el javascript se recomienda la herramienta <http://closure-compiler.appspot.com/home> y el css puede ser minificado con <http://cssminifier.com/>.

También el tamaño de las imágenes es importante. Una imagen puede reducir considerablemente su tamaño sin perder calidad. Una de las mejores herramientas para esta tarea es Photoshop, que nos permite exportar las imágenes para web. En este vídeo podemos observar como guardar los archivos de manera que estos queden optimizados para la web: <http://tv.adobe.com/watch/understanding-adobe-photoshop-cs6/save-for-web/>

Los textos están disponibles bajo la licencia de © 2013 Intelligence Sense en conjunto con Pixel506